Table 10–3. UC07/08 Controller Format Modes

Bits					
04	03	02	01	00	Mode
0	0	0	0	0	Not used.
0	1	0	0	0	**Format Mode.** The target controller performs a complete format of the specified disk and destroys the old Bad Sector File. A new Bad Sector File can be supplied by the user. This mode can be used during the format operation on an uninitialized disk drive.
1	0	0	0	0	Not used.
1	1	0	0	0	**Format Mode with Update.** In this mode, the target uses the Bad Sector File already on the disk and adds to it. This mode can also be used to format an uninitialized disk drive.

Interleave Code - Bytes 03 through 04

The code in the Interleave field requests that the logical blocks be related in a specific fashion to the physical blocks to compensate for differences in execution time between the host processor and the target Controller, if necessary. Emulex recommends that a 1:1 sequential interleave be used for Winchester drives.

The most significant byte of the Interleave field (Byte 03) must be zero. An Interleave value of zero (hexadecimal) requests that the Target use its default interleave.

If a value greater than 1 is specified in the Interleave field, that number indicates where the next logical block in sequence is located with respect to the logical block just before it. For example, if an interleave code of 3 is specified and n is a logical block, then n + 1 is the third contiguous block from n.

10.7.2 Inquiry (Opcode 12H)

The **INQUIRY** CDB, shown below, is a means by which the Initiator can request information regarding the controller and its attached peripheral device.

When this command is sent to a nonexistent LUN, the controller transfers the **INQUIRY** data back to the initiator and terminates the command with a GOOD status. The Initiator examines the Device Type Qualifier field to determine if it is a valid LUN.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	1	0	0	1	0
01	Logical Unit Number			Reserved				
02	Reserved							
03	Reserved							
04	Allocation Length							
05	Vendor Unique		Reserved				Flag	Link

Allocation Length - Byte 04, Bits <07:00>

These bits specify the number of bytes allocated by the Initiator for returned INQUIRY data. A value of 0 means that no data will be transferred to the Initiator and is not considered an error. Any other value indicates the maximum number of bytes to be transferred.

10.7.3 Rewind (Opcode 01H, Tape Only)

The **REWIND** CDB, shown below, allows you to rewind to the beginning of a tape.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	0	0	0	0	1
01	0	0	0	0	0	0	0	Immed
02	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0
04	0	0	0	0	0	0	0	0
05	X	X	0	0	0	0	0	0

Rewind Timing - Byte 01, Bits <00:00>

If the Immed bit equals 0, then the Rewind command will be completed into the Initiator after the entire operation is complete. If the Immed bit equals 1, then the command will be completed into the Initiator upon acceptance of the command.

10.7.4 Mode Select (Opcode 15H)

The **MODE SELECT** CDB, shown below, enables an Initiator to specify device parameters to the target controller. This command is used only during drive formatting operations.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	1	0	1	0	1
01	LUN			PF	0	0	0	0
02	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0
04	Parameter List Length							
05	0	0	0	0	0	0	0	0

Logical Unit Number (LUN) - Byte 01, Bits <07:05>

These bits specify the LUN of the addressed device for this command.

Page Format - Byte 01, Bit <04>

This byte is supported for devices that have implemented the bit. See subsection 10.7.4 for further details.

Parameter List Length - Byte 04

This byte specifies the length in bytes of the Parameter List sent during the Data Out Phase of this command.

10.7.5 Mode Sense (Opcode 1AH)

The **MODE SENSE** CDB, shown below, enables an Initiator to receive device parameters from the target controller. **MODE SENSE** is a complementary command to the **MODE SELECT** command.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	1	1	0	1	0
01	LUN			PF	0	0	0	0
02	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0
04	Parameter List Length							
05	0	0	0	0	0	0	0	0

Logical Unit Number (LUN) - Byte 01, Bits <07:05>

These bits specify the LUN of the addressed device for this command.

Page Format - Byte 01, Bit <04>

This byte is supported for devices that have implemented the bit. See subsection 10.7.4 for further details.

Parameter List Length - Byte 04

This byte specifies the length in bytes of the Parameter List sent during the Data In Phase of this command.

10.7.6 Read (Opcode 08H, Tape)

The **READ** CDB for tape, shown below, causes the target to transfer one or more blocks of data to the Initiator.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	0	1	0	0	0
01	0	0	0	0	0	0	SILI	Fixed
02	Transfer Length (msb)							
03	Transfer Length							
04	Transfer Length (lsb)							
05	X	X	0	0	0	0	0	0

The following definitions clarify bits 0 and 1:

Byte 1 Bit 0 = Fixed
0 = Read Variable Block Mode
1 = Read Fixed Length Mode

Byte 1 Bit 1 = SILI
0 = Do not suppress Illegal Length Check condition status
1 = Suppress Illegal Length Check condition status

10.7.7 Read (08H, Disk)

The **READ** CDB for disk, shown below, causes the transfer of data from the Target device to the Initiator. The **READ** command specifies the starting block number and the number of data blocks to be read. The **READ** command terminates when the number of data blocks to be read is transferred.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	0	1	0	0	0
01	LUN			Logical Block Address (MSB)				
02	Logical Block Address							
03	Logical Block Address (LSB)							
04	Number of Blocks to Transfer (LSB)							
05	0	0	0	0	0	0	0	0

If the disconnect function is enabled, the target controller can disconnect from the Initiator while executing this command.

Logical Unit Number (LUN) - Byte 01, Bits < 07:05 >

These bits specify the LUN of the addressed device for this command.

Logical Block Address - Bytes 01 through 03

These bytes specify the logical block address where the read operation is to begin.

Number of Blocks to Transfer - Byte 04

This byte specifies the number of contiguous logical blocks of data to be transferred. When this byte is zero, 256 logical blocks of data are transferred. Any other value between 1 and 256, inclusive, indicates that number of logical blocks is to be transferred.

10.7.8 Re-Assign Block (Opcode 07H)

The **RE-ASSIGN BLOCK** CDB, shown below, sends a defect block to the Target during the Data Phase of the command. The defect block is a single logical block addresses that is to be re-assigned.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	0	0	1	1	1
01	LUN			0	0	0	0	0
02	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0
04	0	0	0	0	0	0	0	0
05	0	0	0	0	0	0	0	0

Logical Unit Number (LUN) - Byte 01, Bits <07:05>

These bits specify the LUN of the addressed device for this command.

10.7.8.1 Re-assign Block Header

During the Data Out Phase of the **RE-ASSIGN BLOCK** command, data is sent in two pieces. The first piece is the Block Header, shown below, which defines the length of the defect list. Since the UC07/08 re-assigns only a single block at a time, this length is always four bytes.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	0
02	Length of Defect List (MSB)							
03	Length of Defect List (LSB)							

Length of Defect List - Bytes 02 through 03

The value in the Length of Defect List field specifies the total number of bytes (not the total number of defect descriptors) sent during the Data Out Phase of the **RE-ASSIGN BLOCK** command. It is always four.

10.7.8.2 Re-assign Block Defect Descriptor Format

The **RE-ASSIGN BLOCK** defect descriptor is shown below.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	Defect Block Address (MSB)							
01	Defect Block Address							
02	Defect Block Address							
03	Defect Block Address (LSB)							

The Defect Block Address bytes are converted to the appropriate track and block addresses by the target controller. Block addresses that specify previously detected or specified addresses are processed and produce duplicate entries in the Bad Sector File if the re-assignment results in the use of another spare block.

If the LUN has insufficient capacity to re-assign the defective logical block, the target controller terminates the **RE-ASSIGN BLOCK** command with a CHECK CONDITION status code and sets the Sense Key in the Extended Sense Byte to MEDIUM ERROR. The Logical Block Address is returned in the Information Bytes of the sense data.

Defect List Block Address - Bytes 00 through 03

These bytes specify the address of the block that contains the defect.

10.7.9 Request Sense (Opcode 03H)

The **REQUEST SENSE** CDB, shown below, is used to obtain more detailed information after execution of a command. Typically, a **REQUEST SENSE** command is issued after a previous command has completed and a CHECK CONDITION status code has been issued to the Initiator.

An Initiator normally issues a **REQUEST SENSE** command as soon as it receives a CHECK CONDITION status code to obtain the Sense data saved by the target controller.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	0	0	0	1	1
01	LUN			0	0	0	0	0
02	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0
04	Number of Requested Sense Bytes							
05	0	0	0	0	0	0	0	0

Logical Unit Number (LUN) - Byte 01, Bits <07:05>

These bits specify the LUN of the addressed device for this command.

Number of Requested Sense Bytes - Byte 04

This byte contains the number of bytes of data the Initiator has allocated for the sense information. The UC07/08 always allocates 128 bytes for the sense data, although it normally looks only at the first seven bytes. The target controller does not have to return this many bytes. Although the UC07/08 asks for 128 bytes, it will accept any number between 4 and 128.

10.7.9.1 Extended Sense Byte Format

The Extended Sense Byte format is shown below. Note that the format is only seven bytes long, although the controller can actually return up to 128 bytes. If it does, the last 121 bytes will be ignored.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	VADD	1	1	1	0	0	0	0
01	0	0	0	0	0	0	0	0
02	0	0	0	0	Sense Key			
03	Logical Block Address (MSB							
04	Logical Block Address							
05	Logical Block Address							
06	Logical Block Address (LSB)							

Valid Address (VADD) - Byte 00, Bit 07

If this bit is set, the Logical Block Address (Bytes 03 through 06) contains valid information related to the error code.

Sense Key - Byte 02, Bits < 03:00 >

The Sense Key bits indicate status information about any errors detected during the operation. The errors are listed and defined in Table 10–4.

Logical Block Address – Bytes 03 through 06

These bytes specify the Logical Block Address where the error specified by the Sense Key Error Code occurred.

Table 10–4. Extended Sense Byte Sense Keys

Code	Message	Description
00	NO SENSE	There is no Sense Key information to be reported for the designated LUN. This code occurs for a successfully completed command.
01	RECOVERED ERROR	The last command was completed successfully, but with some recovery action performed by the Target.
02	NOT READY	The addressed LUN cannot be accessed. Operator intervention might be required.
03	MEDIUM ERROR	The command terminated with a nonrecoverable-error condition which was probably caused by a flaw in the media or by an error in the recorded data.
04	HARDWARE ERROR	A nonrecoverable hardware error (e.g., controller failure, device failure, parity error, etc.) was detected while the Target was performing the command or while the Target was performing a Self-Test operation.
05	ILLEGAL REQUEST	There was an illegal parameter in the command or in the additional required parameters supplied as data for some related commands.
06	UNIT ATTENTION	The addressed LUN has been reset. This error is reported the first time any command is issued after the condition is detected; then the requested command is not performed. This condition is cleared when the next command is issued by the same host adapter. UNIT ATTENTION is reported to all SCSI devices that subsequently issue a command to the LUN.
07	DATA PROTECT	A write operation was attempted on a write-protected device.
08	BLANK CHECK	A blank sector was encountered during read operation or a previously written sector was encountered during a write operation.
08-FF	DRIVE ERROR	These codes are interpreted by the UC07/08 as nonrecoverable and nonretryable drive errors.

10.7.10 Start/Stop Unit (Opcode 1BH)

The **START/STOP UNIT**, shown below, requests that the target controller cause the LUN to spin up or spin down. The controller automatically performs a **START/STOP UNIT** function during its initialization operation.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	1	1	0	1	1
01	LUN			0	0	0	0	Immed
02	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0
04	0	0	0	0	0	0	0	Start
05	0	0	0	0	0	0	0	0

Logical Unit Number (LUN) – Byte 01, Bits <07:05>

These bits specify the LUN of the addressed device for this command.

Immediate Bit – Byte 01, Bit 00

When set to 1, this bit indicates the status will be returned as soon as the operation is initiated. If the bit is cleared, the controller will return the status when the operation is completed.

Start/Stop Bit – Byte 04, Bit 00

When set to 1, the START bit requests the logical unit be made ready for use. When cleared, the bit requests that the logical unit be stopped.

10.7.11 Test Unit Ready (Opcode 00H)

The **TEST UNIT READY** CDB, shown below, causes a test to be performed to ensure the disk drive is powered-on and ready. This condition is indicated by a GOOD status code being returned in response to this command.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	0	0	0	0	0
01	LUN			0	0	0	0	0
02	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0
04	0	0	0	0	0	0	0	0
05	0	0	0	0	0	0	0	0

Logical Unit Number (LUN) – Byte 01, Bits <07:05>

These bits specify the LUN of the addressed device for this command.

10.7.12 Write Filemarks (Opcode 10H, Tape Only)

The **WRITE FILEMARKS** CDB, shown below, allows you to write a specified number of filemarks.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	1	0	0	0	0
01	0	0	0	0	0	0	0	0
02	Number of Filemarks (msb)							
03	0	0	Number of Filemarks					
04	0	0	Number of Filemarks (lsb)					
05	0	0	0	0	0	0	0	0

10.7.13 Space (Opcode 11H, Tape Only)

The **SPACE** CDB, shown below, is used to position in the forward or reverse direction, over elements of the data set (data blocks, filemarks, occurrence of *n* sequential filemarks, end-of-data).

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	1	0	0	0	1
01	0	0	0	0	0	0	Code	
02	Count (msb)							
03	0	0	Count					
04	0	0	0	Count (lsb)				
05	X	X	0	0	0	0	0	0

Count specifies the number of elements to be spaced over. A positive value of Count spaces in the forward direction. To space in the reverse direction, set Count to the 2's complement of the number of elements to be spaced over.

Code specifies the spacing action, as follows:

Code		Action
0	0	Space Data Blocks
0	1	Space Filemarks

10.7.14 Erase (Opcode 19H, Tape Only)

The ERASE CDB, shown below, allows you to erase a section of tape.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	1	1	0	0	1
01	0	0	0	0	0	0	0	Long
02	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0
04	0	0	0	0	0	0	0	0
05	X	X	0	0	0	0	0	0

If the Long bit equals 1, then a full erase from the current logical position to the physical end of tape will occur.

10.7.15 Load/Unload (Opcode 1BH, Tape Only)

The LOAD/UNLOAD CDB, shown below, allows you to load or unload the tape in the drive.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	1	1	0	1	1
01	0	0	0	0	0	0	0	Immed
02	0	0	0	0	0	0	0	0
03	0	0	0	0	0	0	0	0
04	0	0	0	0	0	0	0	Load
05	X	X	0	0	0	0	0	0

The Load bit specifies the load or unload actions, as follows:

Load	Action
0	Rewind, Unload, drive is NOT READY
1	Load, Rewind to BOT, drive is READY

10.7.16 Write (Opcode 0AH, Tape)

The **WRITE** CDB for tape, shown below, transfers one or more bytes of data from the Initiator to the current position on the tape.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	0	1	0	1	0
01	0	0	0	0	0	0	0	Fixed
02	Transfer Length (msb)							
03	Transfer Length							
04	Transfer Length (lsb)							
05	X	X	0	0	0	0	0	0

10.7.17 Write (Opcode 0AH, Disk)

The **WRITE** CDB for disk, shown below, causes data to be transferred from the Initiator to the Target device. The amount of data written is a multiple of the block length. The **WRITE** command specifies the starting logical block number and the number of blocks to be written.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	0	0	1	0	1	0
01	LUN			Logical Block Address (MSB)				
02	Logical Block Address							
03	Logical Block Address (LSB)							
04	Number of Blocks in Transfer							
05	0	0	0	0	0	0	0	0

If the disconnect function is enabled, the target controller can disconnect from the Initiator while executing this command.

Logical Unit Number (LUN) – Byte 01, Bits <07:05>

These bits specify the LUN of the addressed device for this command.

Logical Block Address – Bytes 01 through 03

These bytes specify the logical block where the write operation is to begin.

Number of Blocks to Transfer – Byte 04

This byte specifies the number of contiguous logical blocks of data to be transferred. When this byte contains all zeros, 256 logical blocks of data are transferred. Any other Number of Blocks to Transfer value indicates that number of blocks are to be transferred.

10.8 SCSI Group Code 1 Command Descriptions

SCSI Group Code 1 command names and operation codes supported by the UC07/08 are listed by subsection number in the following table. These commands will only be used if the device size exceeds the limitation of group 0 command set.

Subsection	UC07/08 SCSI Command	Opcode (Hexadecimal)
10.8.1	READ CAPACITY	25
10.8.2	READ (EXTENDED)	28
10.8.3	WRITE (EXTENDED)	2A

This subsection provides detailed descriptions of the commands, including CDB formats, hexadecimal operation code, byte and bit functions, and any necessary effects produced by the commands. Each UC07/08 SCSI command is described in a separate subsection.

A sample Group 1 CDB is shown in Figure 10–4. The first byte of a command (Byte 00) contains two fields: the Group Code in the high-order three bits (bits <07:05>), and the Command Code in the low-order five bits (bits <04:00>). The Group Code determines the length of the command packet in the CDB, and together the Group and Command Codes determine the operation to be performed. Bits <07:05> of byte 01 in the CDB contain the LUN of the device being addressed.

The last byte (byte 09) in every CDB is a Control Byte which is always zero. The remaining bytes in the CDB contain Command-Dependent Parameters.

NOTE: If a byte in a CDB can be any bit pattern, it is specified as **Not Used** in the paragraphs describing that CDB. If a byte in a CDB must be all zeros, it is specified as **Reserved** in the paragraphs describing that CDB.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	Group Code			Command Code				
01	LUN			Command-Dependent Parameters				
02	Command-Dependent Parameters							
03	Command-Dependent Parameters							
04	Command-Dependent Parameters							
05	Command-Dependent Parameters							
06	Command-Dependent Parameters							
07	Command-Dependent Parameters							
08	Command-Dependent Parameters							
09	0	0	0	0	0	0	0	0

Figure 10–4. Sample Group 1 Command Descriptor Block

10.8.1 Read Capacity (Opcode 25H)

The **READ CAPACITY** CDB, shown below, allows the Initiator to determine the maximum accessible logical block number on the specified LUN. This command also returns the size of the logical block. The information is returned to the Initiator during the Data In phase.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	1	0	0	1	0	1
01	LUN			0	0	0	0	0
02	Logical Block Address (MSB)							
03	Logical Block Address							
04	Logical Block Address							
05	Logical Block Address (LSB)							
06	0	0	0	0	0	0	0	0
07	0	0	0	0	0	0	0	0
08	0	0	0	0	0	0	0	0
09	0	0	0	0	0	0	0	0

Logical Unit Number (LUN) – Byte 01, Bits <07:05>

These bits specify the LUN of the addressed device for this command.

Logical Block Address – Bytes 02 through 05

These bytes specify the logical block address where the Extended read operation is to begin.

Reserved – Bytes 06 through 09

These bytes are reserved and must be zero.

10.8.2 Read (Extended) (Opcode 28H)

The **READ (EXTENDED)** CDB, shown below, performs the same function as the Group 0 READ command: it causes the transfer of data from the Target device to the Initiator. The amount of transferred data is a multiple of the block length (i.e., 512 data bytes/block). The **READ (EXTENDED)** command specifies the starting block number and the number of data blocks to be read. The **READ (EXTENDED)** command terminates when the number of data blocks to be read has been transferred.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	1	0	1	0	0	0
01	LUN			0	0	0	0	0
02	Logical Block Address (MSB)							
03	Logical Block Address							
04	Logical Block Address							
05	Logical Block Address (LSB)							
06	0	0	0	0	0	0	0	0
02	Number of Blocks (MSB)							
02	Number of Blocks (LSB)							
09	0	0	0	0	0	0	0	0

If the disconnect function is enabled, the Target can disconnect from the Initiator while executing this command.

If any reservation access conflict exists, the Target terminates the READ (EXTENDED) command with a RESERVATION CONFLICT status code; no data is read.

If an error occur curing a read operation, the Target terminates the READ (EXTENDED) command, sends a CHECK CONDITION status code to the Initiator, and sets the Sense Key that defines the error condition in the Extended Sense Byte.

Logical Unit Number (LUN) – Byte 01, Bits <07:05>

These bits specify the LUN of the addressed device for this command.

Logical Block Address – Bytes 02 through 05

These bytes specify the logical block address where the Extended read operation is to begin.

Reserved – Byte 06

This byte is reserved and must be zero.

Number of Blocks – Bytes 07 through 08

This byte specifies the number of contiguous logical blocks of data to be transferred. When this byte contains all zeros, no logical blocks of data are transferred. Any other value between 1 and 256, inclusive, indicates that number of logical blocks are to be transferred.

10.8.3 Write (Extended) (Opcode 2AH)

The **WRITE (EXTENDED)** CDB, shown below, performs the same function as the Group 0 **WRITE** command; it causes data to be transferred from the Initiator to the Target device. The amount of data written is a multiple of the block length. The **WRITE (EXTENDED)** command specifies the starting logical block number and the number of blocks to be written.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	0	0	1	0	1	0	1	0
01	LUN			0	0	0	0	0
02	Logical Block Address (MSB)							
03	Logical Block Address							
04	Logical Block Address							
05	Logical Block Address (LSB)							
06	0	0	0	0	0	0	0	0
07	Number of Blocks (MSB)							
08	Number of Blocks (MSB)							
09	0	0	0	0	0	0	0	0

If the disconnect function is enabled, the Target can disconnect from the Initiator while executing this command.

Logical Unit Number (LUN) – Byte 01, Bits <07:05>

These bits specify the LUN of the addressed device for this command.

Logical Block Address – Bytes 02 through 05

These bytes specify the logical block where the Extended write operation is to begin.

Reserved – Byte 06

This byte is reserved and must be zero.

Number of Blocks – Bytes 07 through 08

This byte specifies the number of contiguous logical blocks of data to be transferred. When the Number of Blocks to Transfer is zero, no logical blocks of data are transferred. Any other value between 1 and 256, inclusive, indicates that number of logical blocks are to be transferred.

10.9 SCSI Group Code 7 Command Descriptions

SCSI Group Code 7 commands are vendor-unique SCSI commands. Group code 7 command names and operation codes are issued by the UC07/08. These commands must be supported for MSCP-type replacements.

This subsection provides detailed descriptions of the commands, including CDB formats, hexadecimal operation code, byte and bit functions, and any necessary event-sequence descriptions (i.e., effects produced by the commands). Each SCSI command is described in a separate subsection.

A sample Group 7 CDB is shown in Figure 10-5. The first byte of a command (Byte 00) contains two fields: the Group Code in the high-order three bits (bits <07:05>), and the Command Code in the low-order five bits (bits <04:00>). The Group Code determines the length of the command packet in the CDB, and together the Group and Command Codes determine the operation to be performed.

Bits <07:05> of byte 01 in the CDB contain the LUN of the device being addressed. The UC07/08, acting as a SCSI bus Initiator, supports up to eight LUNs: eight disk drives, assuming that none of the drives are split. The LUN must be specified for all commands. If a LUN value issued by the Initiator in an IDENTIFY message differs from the value specified in the CDB, that value supersedes the value specified in the CDB. The definition of the low-order bits in byte 01 is based on the current command.

The last byte is reserved and is always zero. The remaining bytes in the CDB are primarily command-dependent.

NOTE: Bits 00 and 01 of the last byte are normally used to link commands. However, the UC07/08 never links commands, so these bits are always zero.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	Group Code			Command Code				
01	LUN			Command-Dependent Parameters				
02	Command-Dependent Parameters							
03	Command-Dependent Parameters							
04	Command-Dependent Parameters							
05	Command-Dependent Parameters							
06	Command-Dependent Parameters							
07	Command-Dependent Parameters							
08	Command-Dependent Parameters							
09	0	0	0	0	0	0	0	0

Figure 10-5. Sample Group 7 Command Descriptor Block

10.9.1 Read Long (Opcode E8H)

The **READ LONG** CDB, shown below, causes the target controller to perform a read operation of one data block, beginning at the specified block address. The data and the six Error Correction Code (ECC) bytes of the specified block are transferred to the Initiator. The ECC bytes follow the data.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	1	1	1	0	1	0	0	0
01	LUN			0	0	0	0	0
02	0	Block Address (MSB)						
03	Block Address							
04	Block Address							
05	Block Address (LSB)							
06	0	0	0	0	0	0	0	0
07	0	0	0	0	0	0	0	0
08	0	0	0	0	0	0	0	0
09	0	0	0	0	0	0	0	0

Block Address – Bytes 02 through 05

These bytes specify the block address where the read long operation is to begin.

10.9.2 Write Long (Opcode EAH)

The **WRITE LONG** CDB, shown below, causes the target controller to perform a write operation of one data block, starting at the specified logical block address. The data and the six ECC bytes of the specified logical block are written for each logical block specified in the logical block address. The ECC bytes follow the data.

Byte	Bit 07	Bit 06	Bit 05	Bit 04	Bit 03	Bit 02	Bit 01	Bit 00
00	1	1	1	0	1	0	0	1
01	LUN			0	0	0	0	0
02	0	Block Address (MSB)						
03	Block Address							
04	Block Address							
05	Block Address (LSB)							
06	0	0	0	0	0	0	0	0
07	0	0	0	0	0	0	0	0
08	0	0	0	0	0	0	0	0
09	0	0	0	0	0	0	0	0

Block Address – Bytes 02 through 05

These bytes specify the block address where the write long operation is to begin.

NOTE: The device must not change the ECC bytes supplied by the UC07/08. These bytes must be written on the block specified by the command.

Appendix A
AUTOCONFIGURE, CSR AND VECTOR ADDRESS

A.1 Overview

The following discussion presents the algorithm for assignment of floating addresses and vectors for all DEC operating systems. Bus addresses are discussed in subsection 4.3.2.

A.2 Determining the CSR Address for use with Autoconfigure

The term Autoconfigure refers to a software utility that is run when the computer is bootstrapped. This utility finds and identifies I/O devices in the I/O page of system memory.

Some devices (like the DM11) have fixed addresses reserved for them. Autoconfigure detects their presence by simply testing their standard address for a response. Specifically, the control/status register (CSR) address, which is usually the first register of the block, is tested.

Addresses for those devices not assigned fixed numbers are selected from the floating CSR address space of the Q-Bus input/output (I/O) page. This means that the presence or absence of floating devices will affect the assignment of addresses to other floating-address devices. Similarly, many devices have floating interrupt vector addresses. According to the DEC standard, vectors must be assigned in a specific sequence and the presence of one type of device will affect the correct assignment of vectors for other devices.

The CSR address for a floating-address device is selected according to the algorithm used during autoconfigure. The algorithm is used in conjunction with a Device Table, Table A-1.

Essentially, Autoconfigure checks each valid CSR address in the floating CSR address space for the presence of a device. Autoconfigure expects any devices installed in that space to be in the order specified by the Device Table. Also, the utility expects an eight-byte block to be reserved for each device that is not installed in the system. Each empty block tells Autoconfigure to look at the next valid address for the next device on the list.

When a device is detected, a block of addresses is reserved for the device according to the number of registers it employs. The utility then looks at the next CSR for that device type. If there is a device there, it is assumed to be of the same type as the one before it and a block is reserved for that device. If there is no response at the next address, that space is reserved to indicate that there are no more devices of that type. Then the utility checks the CSR address (at the appropriate boundary) for the next device in the table.

Table A–1. SYSGEN Device Table

Rank	Device	Number of Registers	Octal Modulus	Rank	Device	Number of Registers	Octal Modulus
1	DJ11	4	10	17	Reserved	4	10
2	DH11	8	20	18	RX11[2]	4	10
3	DQ11	4	10	18	RX211[2]	4	10
4	DU11, DUV11	4	10	18	RXV11[2]	4	10
5	DUP11	4	10	18	RXV21[2]	4	10
6	LK11A	4	10	19	DR11-W	4	10
7	DMC11	4	10	20	DR11-B[3]	4	10
7	DMR11	4	10	21	DMP11	4	10
8	DZ11[1]	4	10	22	DPV11	4	10
8	DZV11	4	10	23	ISB11	4	10
8	DZS11	4	10	24	DMV11	8	10
8	DZ32	4	10	25	DEUNA[2]	4	10
9	KMC11	4	10	**26**	**UDA50[2]**	**2**	**4**
10	LPP11	4	10	27	DMF32	16	40
11	VMV21	4	10	28	DMS11	6	20
12	VMV31	8	20	29	VS100	8	20
13	DWR70	4	10	**30**	**TU81**	**2**	**4**
14	RL11[2]	4	10	31	KMV11	8	20
14	RLV11[2]	4	10	32	DHV11	8	20
15	LAP11-K[2]	8	20	33	DMZ32	16	40
16	KW11-C	4	10	34	CP132	16	40

[1] DZ11-E and DZ11-F are treated as two DZ11s.

[2] The first device of this type has a fixed address. Any extra devices have a floating address.

[3] The first of two devices of this type have a fixed address. Any extra devices have a floating address.

In summary, there are four rules that pertain to the assignment of device addresses in floating address space:

1. Devices with floating addresses must be attached in the order in which they are listed in the Device Table, Table A–1.
2. The CSR address for a given device type is assigned on word boundaries according to the number of Q-Bus-accessible registers that the device has. The following table relates the number of device registers to possible word boundaries.

Device Registers	Possible Boundaries
1	Any Word
2	XXXXX0, XXXXX4
3, 4	XXXXX0
5, 6, 7, 8	XXXX00, XXXX20, XXXX40, XXXX60
9 through 16	XXXX00, XXXX40

The Autoconfigure utility inspects for a given device type only at one of the possible boundaries for that device. That is, the utility does not look for a DMF32 (16 registers) at an address that ends in 20.

3. An 8-byte gap must follow the register block of any installed device to indicate that there are no more of that type of device. This gap must start on the proper CSR address boundary for that type of device.
4. An 8-byte gap must be reserved in floating address space for each device type that is not installed in the current system. The gap must start on the proper word boundary for the type of device the gap represents. That is, a single DJ11 installed at 760010 would be followed by a gap starting at 760020 to show a change of device types. A gap to show that there are none of the next device on the list, a DH11, would begin at 760040, the next legal boundary for a DH11-type device.

A.3 Determining the Vector Address for use with Autoconfigure

There is a floating vector address convention that is used for communications and other devices which interface with the Q-Bus. These vector addresses are assigned in order starting at 300 and proceeding upwards to 777. Table A–2 shows the assignment sequence. For a given system configuration, the device with the highest floating vector rank would be assigned to vector address 300. Additional devices of the same type would be assigned subsequent vector addresses according to the number of vectors required per device, and according to the starting boundary assigned to that device type.

Table A–2. Priority Ranking for Floating Vector Addressing (starting at 300_8 and proceeding upwards)

Rank	Device	Number of Vectors	Octal Modulus
1	DC11	2	10
1	TU58	2	10
2	KL11[1]	2	10
2	DL11-A[1]	2	10
2	DL11-B[1]	2	10
2	DLV11-J[1]	8	40
2	DLV11, DLV11-F[1]	2	10
3	DP11	2	10
4	DM11-A	2	10
5	DN11	1	4
6	DM11-BB/BA	1	4
7	DH11 modem control	1	4
8	DR11-A, DRV11-B	2	10
9	DR11-C, DRV11	2	10
10	PA611 (reader + punch)	4	20
11	LPD11	2	10
12	DT07	2	10
13	DX11	2	10
14	DL11-C to DLV11-F	2	10
15	DJ11	2	10
16	DH11	2	10

(Continued on next page)

Table A–2. Priority Ranking for Floating Vector Addressing (starting at 300_8 and proceeding upwards)(Continued)

Rank	Device	Number of Vectors	Octal Modulus
17	VT40	4	20
17	VSV11	4	10
18	LPS11	6	40
19	DQ11	2	10
20	KW11-W, KWV11	2	10
21	DU11, DUV11	2	10
22	DUP11	2	10
23	DV11 + modem control	3	20
24	LK11-A	2	10
25	DWUN	2	10
26	DMC11	2	10
26	DMR11	2	10
27	DZ11/DZS11/DZV11	2	10
27	DZ32	2	10
28	KMC11	2	10
29	LPP11	2	10
30	VMV21	2	10
31	VMV31	2	10
32	VTV01	2	10
33	DWR70	2	10
34	RL11/RLV11[2]	1	4
35	TS11[2], TU80[2]	1	4
36	LPA11-K	2	10
37	IP11/IP300[2]	1	4
38	KW11-C	2	10
39	RX11[2]	1	4
39	RX211[2]	1	4
39	RXV11[2]	1	4
39	RXV21[2]	1	4
40	DR11-W	1	4

(Continued on next page)

Table A–2. Priority Ranking for Floating Vector Addressing (starting at 300_8 and proceeding upwards)(Continued)

Rank	Device	Number of Vectors	Octal Modulus
41	DR11-B[2]	1	4
42	DMP11	2	10
43	DPV11	2	10
44	ML11[3]	1	4
45	ISB11	2	10
46	DMV11	2	10
47	DEUNA[2]	1	4
48	UDA50[2]	1	4
49	DMF32	8	40
50	KMS11	3	20
51	PCL11-B	2	10
52	VS100	1	4
53	TU81[2]	1	4
54	KMV11	2	10
55	Reserved	2	10
56	IEX	2	10
57	DHV11	2	10
58	DMZ32	6	20
59	CP132	6	20

[1] A KL11 or DL11 used as a console, has a fixed vector.

[2] The first device of this type has a fixed vector. Any extra devices have a floating vector.

[3] ML11 is a Massbuss which can connect to a Q-Bus via a bus adapter.

Vector addresses are assigned on the boundaries indicated in the modulus column of Table A–2. That is, if the modulus is 10, then the first vector address for that device must end with zero (XX0). If the modulus is 4, then the first vector address can end with zero or 4 (XX0, XX4).

Vector addresses always fall on modulo 4 boundaries (XX0, XX4). That is, a vector address never ends in any number but four or zero. Consequently, if a device has two vectors and the first must start on a modulo 10 boundary, then, using 350 as a starting point, the vectors will be 350 and 354.

A.4 A System Configuration Example

Table A–3 contains an example of a system configuration that includes devices with fixed addresses and vectors, and floating addresses and/or vectors.

Table A–4 shows how the device addresses for the floating address devices in Table A–3 were computed, including gaps.

Table A–3. CSR and Vector Address Example

Controller	Vector	CSR
1 UDA50	154	772150
1 DZ11	300	760100
1 UDA50	310	760354
2 DHV11	320, 330	760500, 760500

Table A–4. Floating CSR Address Assignment Example

Installed	Device		Octal Address
	DJ11	Gap	760010
	DH11	Gap	760020
	DQ11	Gap	760030
	DU11	Gap	760040
	DUP11	Gap	760050
	LK11A	Gap	760060
	DMC11	Gap	760070
——>	DZ11		760100
		Gap	760110
	KMC11	Gap	760120
	LPP11	Gap	760130
	VMV21	Gap	760140
	VMV31	Gap	760150
	DWR70	Gap	760170
	RL11	Gap	760200
	LPA11-K	Gap	760220

(Continued on next page)

Table A–4. Floating CSR Address Assignment Example(Continued)

	KW11-C	Gap	760230
	Reserved	Gap	760240
	RX11	Gap	760250
	DR11-W	Gap	760260
	DR11-B	Gap	760270
	DMP11	Gap	760300
	DPV11	Gap	760310
	ISB11	Gap	760320
	DMV11	Gap	760340
	DEUNA	Gap	760350
------>	**UDA50 (UC07/08)**		**772150**
------>	**UDA50 (UC07/08)**		**760354**
		Gap	760360
	DMF32	Gap	760400
	KMS11	Gap	760420
	VS100	Gap	761440
	TU81	**Gap**	**774500**[1]
	TU81	**Gap**	**760444**
	KMV11	Gap	761460
	DHV11		761500
	DHV11		761520
		Gap	761530
	DMZ32	Gap	761540
	CP132	Gap	761600

[1] Fixed Address

Appendix B
PROM REMOVAL AND REPLACEMENT

B.1 Overview

This appendix provides instructions for replacing the firmware PROMs on the UC07 and UC08.

B.2 Exchanging PROMS

The UC07 firmware PROM is located in the socket at U29; the UC08 firmware PROMs are located at U22 and U29. Remove the existing PROMs from their sockets using an IC puller or an equivalent tool.

The PROMs are identified by the part numbers on top of the PROMs. Refer to Figure B–1 for proper pin 1 orientation and place the UC07 PROM in U29 and the UC08 PROMs in U22 and U29. Make certain that each PROM is firmly seated and that no pins are bent or misaligned. (If the two rows of PROM pins are too far apart to fit in the socket, grasp the PROM at its ends using your thumb and forefinger and bend one of the pin rows inward by pressing it against a table top or other flat surface.)

Host Adpater	PROM Number	PCBA Location
UC07	G143	U29
UC08	G143	U22, U29

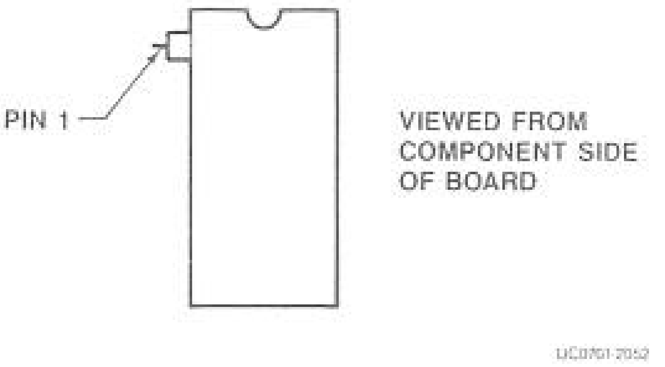

Figure B–1. IC Pin 1 Orientation

Appendix C
DRIVE PARAMETERS

C.1 UC04 Optical Drive Compatibility

Table C–1 lists the optical disk drive NOVRAM parameters for various optical drives. If your application requires you to maintain compatibility with an Emulex UC04 optical disk subsystem, program the UC07/08 NOVRAM using the parameters in this table.

Table C–1. Optical Drive NOVRAM Settings for UC04 Compatibility

	Optimem	Optotech	OSI	Toshiba
Block size	1024	512	1024	1024
Blocks per track	25	20	32	11
Blocks per disk	999,999	390,000	1,000,000	262,723
NOTE: All values are in decimal				

C.2 UC04 Magnetic Drive Compatibility

If your application requires you to maintain format compatibility with an Emulex UC04 magnetic disk subsystem, program the UC07/08 NOVRAM for Type Code 1 or 2, and use the parameter values specified in the UC04 manual.

C.3 General Drive Parameter Settings

Tables C–2 and C–3 list the NOVRAM parameters for some Emulex bridge controller and drive combinations that Emulex has tested. If your drive does not appear in either table, or is an embedded SCSI drive, all the information needed, with the exception of spares and sector offset, can be found in your drive's manual.

If you have an embedded SCSI drive and the manual for the drive does not state the number of sectors per track, you can make the following calculation to determine the value to be entered when F.R.D. asks for the number of physical sectors per track. If the result of the calculation is a fraction, truncate the result to the nearest integer.

$$\text{Sectors per track} = \frac{\text{Physical blocks per drive}}{\text{Physical cylinders per drive X Physical heads per drive}}$$

Example: For a 760M byte drive, having 1624 cylinders, 15 heads, and 1,484,375 physical blocks, how many sectors are there per track?

$$\text{Sectors per track} = \frac{1{,}484{,}375}{1624 \text{ X } 15} = \frac{1{,}484{,}375}{24{,}360} = 60.935$$

which truncates to 60 sectors per track.

If the drive is being formatted with the default values and you get a message during formatting that a mode select error has occurred, you can ignore the message. The value calculated for user size will be close enough that no problem should occur due to slight differences between the actual user size and the calculated user size.

Table C–2. Drive Configuration Parameter Values for Emulex MD01-Based Subsystems

Parameter	**Drive Name**						
	Atasi 3046	**CDC Wren I**	**CDC Wren II–917**	**CDC Wren II–925**	**Fujitsu M2243AS**	**Maxtor XT1140**	**Rodime 208**
Type Code	1	1	1	1	1	1	1
Physical Sectors per Track	17	17	17	17	17	17	17
Heads	7	5	9	9	11	15	8
Physical Cylinders	645	697	917	925	754	918	640
Spare Sectors per Track	1	1	1	1	1	1	1
Spare Cylinders	2	2	2	2	2	2	2
Reduced Write Current	320	697	917	925	754	918	640
Write Precompensation	320	0	0	0	754	918	640
Step Code	1	1	1	1	1	1	1
Removable Media Flag	N	N	N	N	N	N	N

Table C–3. ESDI Drive Configuration Parameter Values for Emulex MD21, MD23, and MD25-Based Subsystems

Parameter	Drive Name								
	CDC Wren III	Fujitsu 2246E	Hitachi DK512-17	Maxtor Ext-4175	Maxtor Ext-4380	Maxtor Ext-8380	Micropolis 1350	Micropolis 1558	Siemens 1300
Type Code	6	6	6	6	6	6	6	6	6
Physical Sectors per Track	36	35	36	34	34	52	35	35	35
Heads	9	10	10	7	15	8	8	15	12
Physical Cylinders	969	823	823	1224	1224	1632	1024	1224	1217
Spare Sectors per Track	1	1	1	1	1	1	1	1	1
Spare Cylinders	2	2	2	2	2	2	2	2	2
Removable Media Flag	N	N	N	N	N	N	N	N	N

Appendix D
TERMINATION POWER

D.1 Overview

This appendix outlines the different methods for providing termination power.

D.2 Termination

SCSI buses are terminated at each end. The UC06/07/08 provides termination for one end of the cable. The last SCSI device on the bus must likewise terminate the bus; however, controllers daisy chained between the UC06/07/08 and last controller on the bus must have their terminators removed. See the controller or subsystem manufacturer's documentation for instructions in installing or removing terminators.

D.2.1 Termination Power

The new dual-wide version of the UC06 and UC07 provides four methods of providing for termination power. Table D-1 shows the jumper setting for each method. Figures D-1 through D-4 illustrate the schematics for termination power.

The dual-wide UC06/07 were designed to comply with the latest SCSI specification concerning SCSI termination power. As configured from the factory, the SCSI terminating network on the UC06/07 uses +5V supplied on pin 26 of the SCSI bus. The 5 volts can be supplied by other devices on the SCSI bus, or by the UC06/07 itself. In the factory configuration, the UC06/07 supplies power to pin 26 of the SCSI bus.

In some cases, the user might require that the UC06/07 provide power for its own terminators without applying power to pin 26. This can be accomplished by cutting one etch and adding a jumper. See Table D-1 for further details.

For the UC06/07 to supply termination power to pin 26, install jumper D-E for the UC07 or JP2 for the UC06. The UC06/07 SCSI port is equipped with a 4.7 mF tantalum capacitor and a .01 mF capacitor. These capacitors prevent noise due to high current switching from affecting termination power.

The termination power on the UC06/07 has the following characteristics:

- 4.40 volts DC to 4.90 volts DC
- Fused at 1 ampere (Fuse: F1) or self–resetable circuit protection
- Diode to prevent backflow of power into the UC07
- Jumpers to enable or disable this feature

Table D–1. Termination Power Options

Setting	Jumper UC06	 UC07	Description
Fact.	JP2 JP1 1–2 JP1 2–3	D–E IN A–B IN B–C OUT	The UC06/07 supplies termination power to the bus. The UC07 termination network is connected to pin 26.
Alt. 1	JP2 JP1 1–2 JP1 2–3	D–E OUT A–B IN B–C OUT	The UC06/07 termination network is connected to pin 26. Pin 26 is unpowered by the UC06/07. Some other device must supply termination power.
Alt. 2	JP2 JP1 1–2 JP1 2–3	D–E IN A–B CUT B–C IN	The UC06/07 supplies termination power to the bus. The UC06/07 termination network is not fused, but is connected to the +5V supplying the UC06/07.
Alt. 3	JP2 JP1 1–2 JP1 2–3	D–E OUT A–B CUT B–C IN	The UC06/07 termination network is connected to the +5V supplying the UC06/07 with no fuse protection. Pin 26 is unpowered by the UC06/07.

NOTE: Jumpers A, B, and C are solder pads located on the board. You must cut the etch on the solder side and install a small wire jumper (component side) for any changes.

D.2.1.1 SCSI Bus Noise Sensitivity

In it's present form, the SCSI bus is very sensitive to noise on termination power. The factory configuration should present no problems in most applications; however, in some applications involving multiple CPUs and a large number of heavily loaded SCSI disks on the same SCSI bus, termination power noise might be detected. If you have an application as described above and experience problems such as data errors, drives going offline, etc., change to Termination Power Alternate 3. If this does not help, contact Emulex Technical Support.

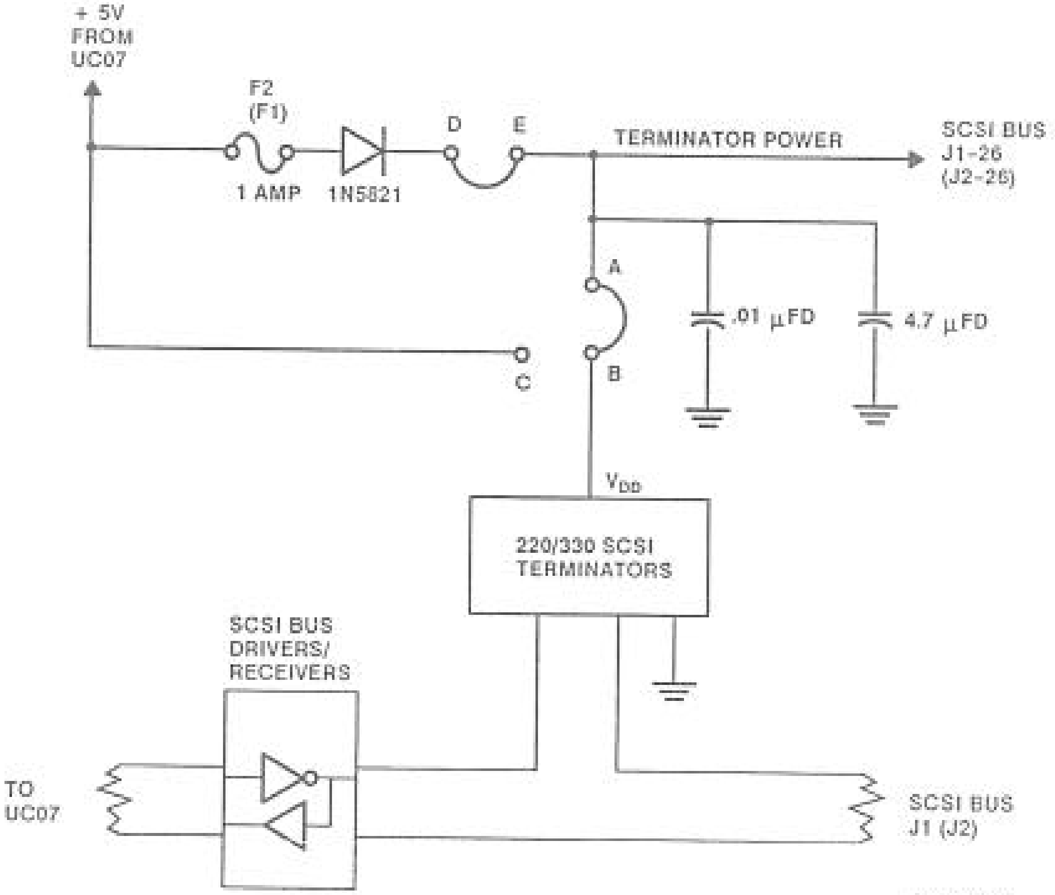

Figure D–1. UC06/07/08 Termination Power Schematic (Factory Configuration)

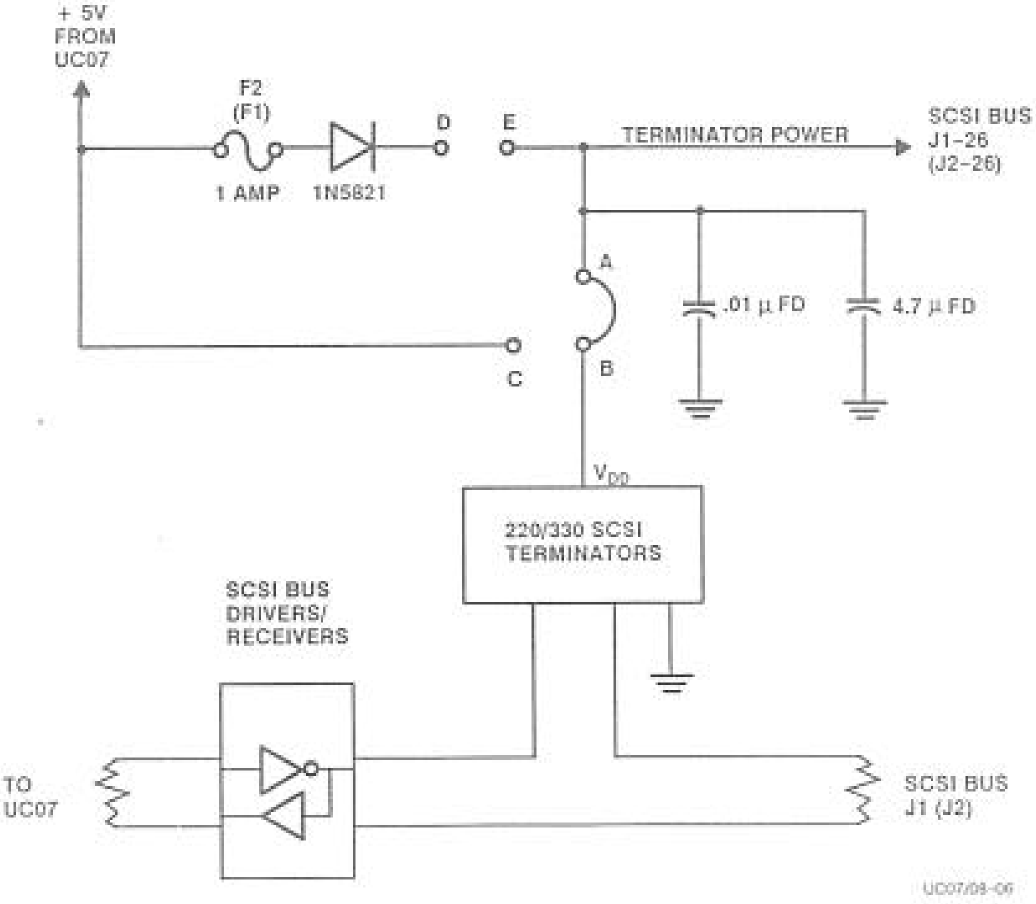

Figure D–2. UC06/07/08 Termination Power Schematic (Alternate 1)

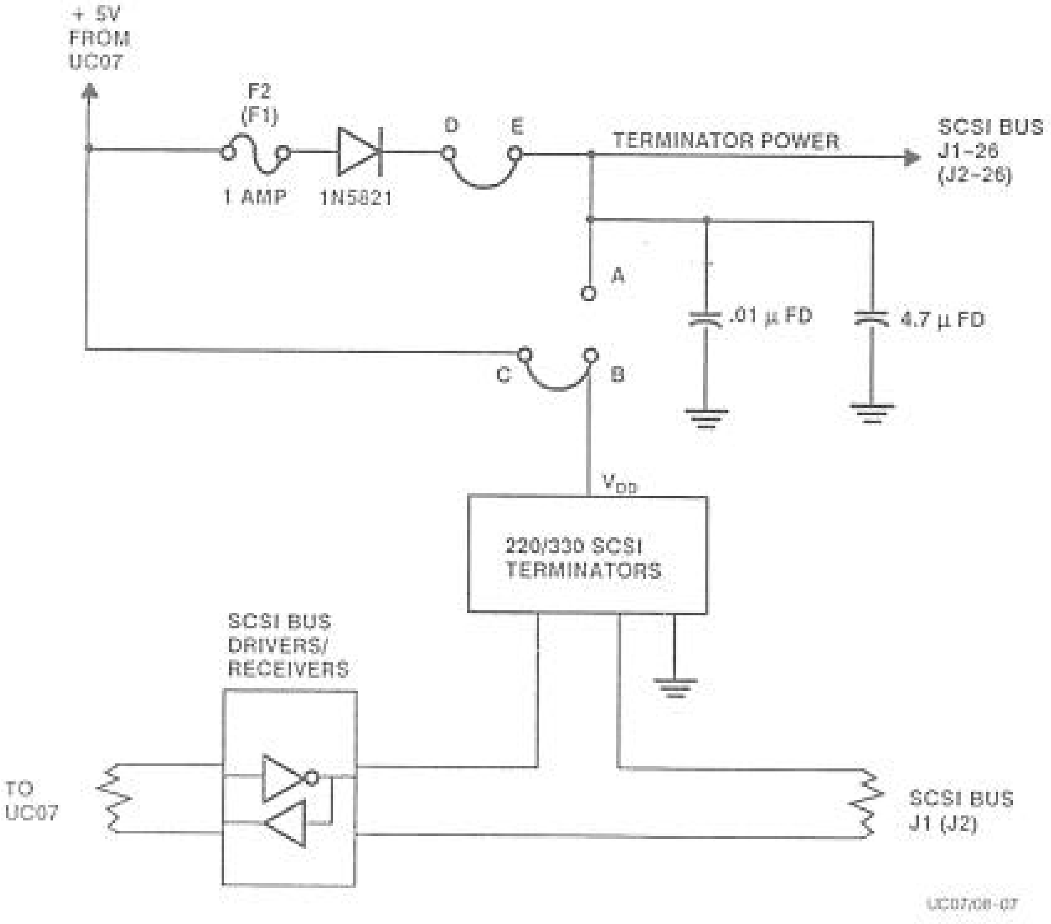

Figure D–3. UC06/07/08 Termination Power Schematic (Alternate 2)

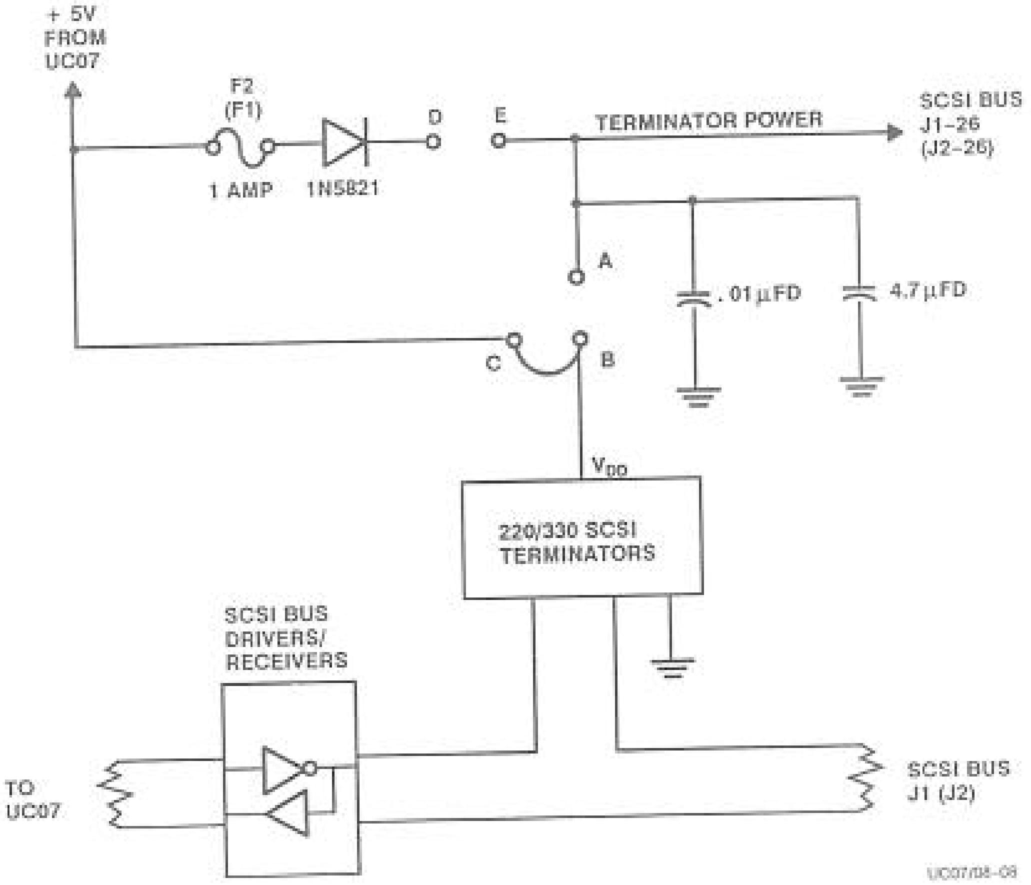

Figure D–4. UC06/07/08 Termination Power Schematic (Alternate 3)